Recent Innovations in Mechatronics (RIiM) Vol. 3. (2016). No. 1-2.
DOI: 10.17667/riim.2016.1-2/15.

DC Motor Simulation with ARM Based Hardware in the Loop

Sándor Csikós
University of Szeged Faculty ofEngineering
TechnicalInstitute
Szeged, Hungary
csikos-s@mk.u-szeged.hu

Ádám Bálint
University of Szeged Faculty of Engineering
Technical Institute
Szeged, Hungary

***Abstract*—Hardware in the loop testing is increasingly important product testing for cutting down time to market. In my previous article I identified the parameters of a DC motor control system using system identification. An implementation of the resulting transfer function was developed on an ARM based hardware in the loop system and has been verified against the original system. The results show that the hardware in the loop system produces responses within tolerance to the stimulus signals and can be used for testing of controlsystems.**

***Keywords*—Hardware in the loop, LabVIEW, ARM microcontroller**

I. Introduction

Model based design is a methodology used in designing and testing controller software implemented in PLCs or microcontrollers. The three phases that phases of model based design are model in the loop (MIL) where only the mathematical representations of the plant and control system are used, software in the loop (SIL) where a simulated implementation of the controller controls a simulation of the plant and hardware in the loop (HIL) where a real controller controls a real-time simulation of a plant. As the last simulation stage of product development hardware in the loop simulation is used to cut down testing time, ensure the correctness of the implemented control algorithm and decrease testing costs. As such it is important to make the simulating device be as close to the real system as possible. Current applications use real-time operating systems to ensure the high refresh rate needed for such simulations and are usually quite costly such as the dSPACE simulator or the National Instruments PXI systems just to name afew.

Real-time computing is divided into soft real-time, where the usefulness of a result degrades after its deadline, firm real-time where infrequent deadline misses are tolerated and hard real-time where missing a deadline is total system failure. Missing deadlines degrades the systems quality therefore a hard real-time simulation method would give us the best results. There are numerous real-time simulations (RTS) currently in literature [1-6], however these systems are soft real-time. Similar research has been conducted by [7] reaching simulation step times of 250-50 μs with a low cost hardware implementation. In this paper experiments were concluded to see if a low cost implementation using an ARM processor could achieve the same results for simple systems, thus furtherlowering the testing cost of testing, also providing an opportunity to create a system for the average user. Such a system could also be used in education due to the low price and strong connection to the current control systems course material being taught at the University ofSzeged.

The desired end result would be to create a framework that enables hard real-time hardware in the loop testing for linear time invariant MIMO systems and systems with nonlinear characteristics such as the pneumatic artificial muscle (PAM) since the other main research topic at the University of Szeged Faculty of Engineering is the high accuracy positioning of pneumatic artificial muscles for robotics and rehabilitation applications [8].

II. Methodology

The hypothesis being tested was that a 32bit ARM microcontroller could be used for hard real-time hardware in the loop simulation. The microcontroller chosen was the STM32F746ZGT6 on the STM32F746 Nucleo board (Fig. 1).

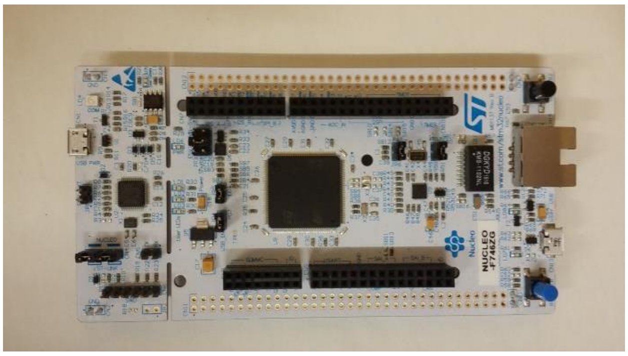

Fig. 1. STM32F746 Nucleo board

The STM32F746ZGT6 features a 216 MHz ARM processor with a built in single precision floating point unit enabling operations on numbers represented as floats to be completed under a couple of cycles depending on the operation. A floating point unit is of key importance as it will be shown.

The programming environment used was the Keil MDK with the STM32CubeMX for pin setup and initialization.Measurements were done using LabVIEW, the

DOI: 10.17667/riim.2016.1-2/15.

data acquisition device used was a myRIO (Fig. 2) from National Instruments.

Fig. 2. myRIO from National Instruments

To test the hypothesis a physical system was constructed consisting of a small DC motor with a magnet mounted on the rotary shaft and an AS5145 absolute magnetic rotary encoder with 12 bits of resolution mounted perpendicular to the shaft (Fig. 3).

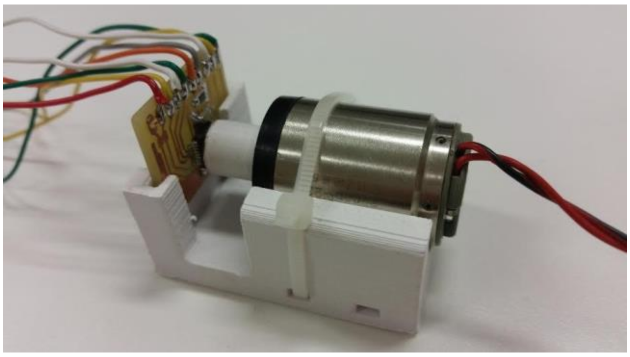
Fig. 3. System to be simulated.DC motor with magnet and AS5145 magneticencoder.

The procedure used was the following. Assuming the model of the system to be the one shown on Fig. 4 where the input of the system is voltage and the output is the rotational speed of the shaft equation (1) and (2) can be derived.

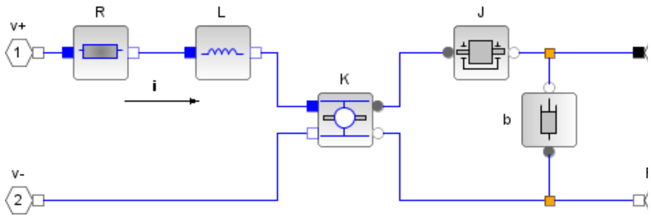

Fig. 4. Block diagram of motor

where:

- v – voltage difference between v+ and v- inV
- i– current flowing through the motor inA
- R – resistance of the motor inΩ
- L – inductance of the motor inH
- K – constant of proportionality inV/rad/s
- J – inertia inkgm^2
- b – rotational dampening constantNms/rad
- $\dot{\theta}$–rotorspeedinrad/s

From the block diagram equation (1) and (2) can be derived.

$$J\frac{d^2\theta}{dt^2} + b\frac{d\theta}{dt} = Ki \quad (1)$$

$$L\frac{di}{dt} + Ri = v - K\frac{d\theta}{dt} \quad (2)$$

Knowing equation (1) and (2) we can apply the Laplace transform to them. By expressing the current we can substitute (1) in (2) and express the transfer function of the motor (3) where the voltage is the input and rotor speed is the output.

$$\hat{g}(s) = \frac{\hat{\dot{\theta}}(s)}{\hat{v}(s)} = \frac{K}{(JLs^2 + (RJ + Lb)s + Rb + K^2)} \quad (3)$$

Knowing the nominator and denominator orders of the transfer function polynomials enables the parametric system identification of the system once a step input of 1 V has been applied and its response measured. Since the rotary encoder only gives out absolute position the angular velocity had to be calculated. To calculate the angular velocity an FPGA program had been written that runs periodically every 400 µs. The results obtained and the result of the system identification can be seen on Fig. 5.

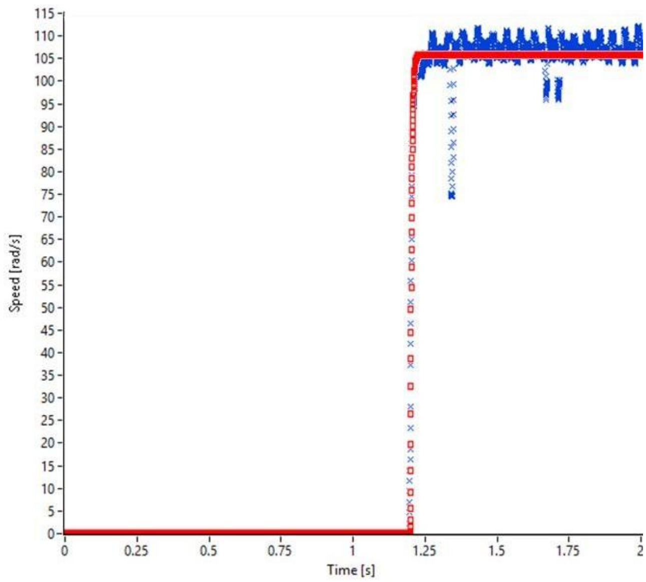

Fig. 5. Results of measurement (blue) and system identification (red)

The transfer function from system identification can be seen in equation (4)

Recent Innovations in Mechatronics (RIiM) Vol. 3. (2016). No. 1-2.
DOI: 10.17667/riim.2016.1-2/15.

$$\hat{g}(s) = \frac{\hat{\theta}(s)}{\hat{v}(s)} = \frac{103.27}{(3.4821 * 10^{-6}s^2 + 0.00519891s + 1)} \quad (4)$$

To implement the acquired transfer function on the microcontroller a bilinear transformation was performed with the sampling frequency of 100kHz to transform it from the s plane to the z plane giving us equation (5) with the following coefficients.

$$\hat{g}(s) = \frac{\hat{\theta}(s)}{\hat{v}(s)} = \frac{103.27}{(3.4821 * 10^{-6}s^2 + 0.00519891s + 1)} \quad (5)$$

Using an inverse z-transform the difference equation was obtained from the discrete time transfer function. Implementing the discrete time transfer function is a simple matter. Since the equation only contains addition multiplication and subtraction and each of these operations only requires 1-3 cycles of the ARMmicrocontroller.

III. Results

Fig. 6 shows a measurement of a step input applied to the implemented hardware in the loop system in contrast with a pure mathematical model.

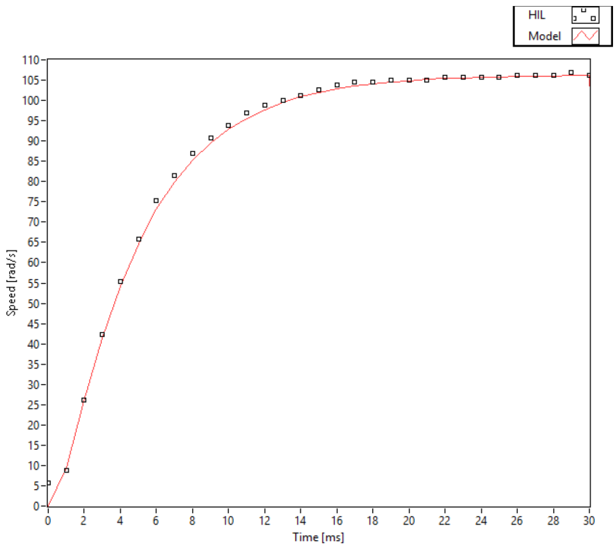

Fig. 6. Results of step input applied to hardware in the loop (squares) and continuous time system model (red)

From the comparison it is visible that the two responsesare close to each other, with an average absolute error of0.88004 rad/s we concluded that using an ARMmicrocontroller is viable for hard real-time hardware in theloop simulation. The STM32F746ZGT6 has a 12 bit analoginput with a maximum sampling frequency of 2.4 MHz thusplacing the maximum sampling frequency of the bilineartransformation at 2.4 MHz, in triple interleave mode thesampling frequency can be boosted to 7.2 MHz, doing soreduces the capabilities of the microcontroller to a single inputsystem.

References

[1] S. Lentijo, A. Monti, E. Santi, C. Welch, and R. Dougal, “A new testing tool for power electronic digital control,” in Proc. IEEE PESC, Jun. 2003, vol. 1, pp. 107–111.

[2] V. Dinavahi, R. Iravani, and R. Bonert, “Design of a real-time digital simulator for a D-STATCOM system,” IEEE Trans. Ind. Electron., vol. 51, no. 5, pp. 1001–1008, Jun. 2004.

[3] H. Li, M. Steurer, K. Shi, S. Woodruff, and D. Zhang, “Development of a unified design, test, and research platform for wind energy systems based on hardware-in-the-loop real-time simulation,” IEEE Trans. Ind. Electron.,vol.53,no.4,pp.1144–1151,Jun.2006.

[4] F. Piazza, S. Squartini, R. Toppi, M. Navarri, M. Pontillo, F. Bettarelli, and A. Lattanzi, “Industry-oriented software-based system for quality evaluation of vehicle audio environments,” IEEE Trans. Ind. Electron., vol. 53, no. 3, pp. 855–866, Jun.2006.

[5] P. Baracos, G. Murere, C. Rabbath, and W. Jin, “Enabling pc-based HIL simulation for automotive applications,” in Proc. IEEE IEMDC, Jun. 2001, pp. 721–729.

[6] M. Park and I. Yu, “A novel real-time simulation technique of photovoltaic generation systems using RTDS,” IEEE Trans. Energy Convers.,vol.19,no.1,pp.164–169,Mar.2004.

[7] Lu, Bin, Xin Wu, Hernan Figueroa, and Antonello Monti. "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls." IEEE Transactions on Industrial Electronics 54, no. 2 (2007): 919-931.

[8] Sárosi J.: “Accurate Positioning of Humanoid Upper Arm” International Journal of Engineering, Annals of Faculty of Engineering Hunedoara, 2011, Vol. 9, No. Extra, ISSN 1584-2673, pp. 33-36